

Colorbit SDK

Development Guide for Applications That Use Colorbit

2012/05/14

Version 1.1.3

**The content of this document is highly confidential
and should be handled accordingly.**

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

1	Introduction	4
2	Reading Colorbits	5
2.1	Colorbit Read Flow	5
2.1.1	Setting the Input Image	5
2.1.2	Separating Colors	5
2.1.3	Extracting Colors	6
2.2	Colorbits That Are Easy to Read	7
2.3	Camera Environment.....	7
2.3.1	Distance From the Camera to the Colorbit.....	7
2.3.2	Ambient Light Brightness	7
2.3.3	Ambient Light Color	7
3	Creating Colorbit	8
3.1	Encoding.....	8
3.1.1	Creating with List Data	8
3.1.2	Creating with the Adobe Illustrator Script	8
3.2	Colorbit Samples	11
	Revision History	13

Figures

Figure 2-1	Example of Colors With Hues Near Red, Green, and Blue	6
Figure 2-2	Example of Red, Green, and Blue Intermediate Colors	6
Figure 3-1	Script Execution Screen 1: Creating Paths	9
Figure 3-2	Script Execution Screen 2: Selecting Paths.....	9
Figure 3-3	Script Execution Screen 3: Setting IDs	10
Figure 3-4	Script Execution Screen 4: Encoding Results.....	10
Figure 3-5	Colorbit Sample 1	11
Figure 3-5	Colorbit Sample 2.....	11
Figure 3-5	Colorbit Sample 3.....	12
Figure 3-5	Colorbit Sample 4.....	12
Figure 3-5	Colorbit Sample 5.....	12

1 Introduction

This development guide describes the Colorbit SDK that can be used for auxiliary markers in Augmented Reality (AR) applications. In order to allow for easy reading from dark environments to light environments without losing the design, recognition is made with 3 colors and 15 cells.

2 Reading Colorbits

This chapter describes the Colorbit read flow, characteristics of Colorbit that are easy to read, and the camera environment.

2.1 Colorbit Read Flow

When reading Colorbit, processing with the Colorbit SDK is performed in the following order: setting the input image, separating colors, and then extracting colors.

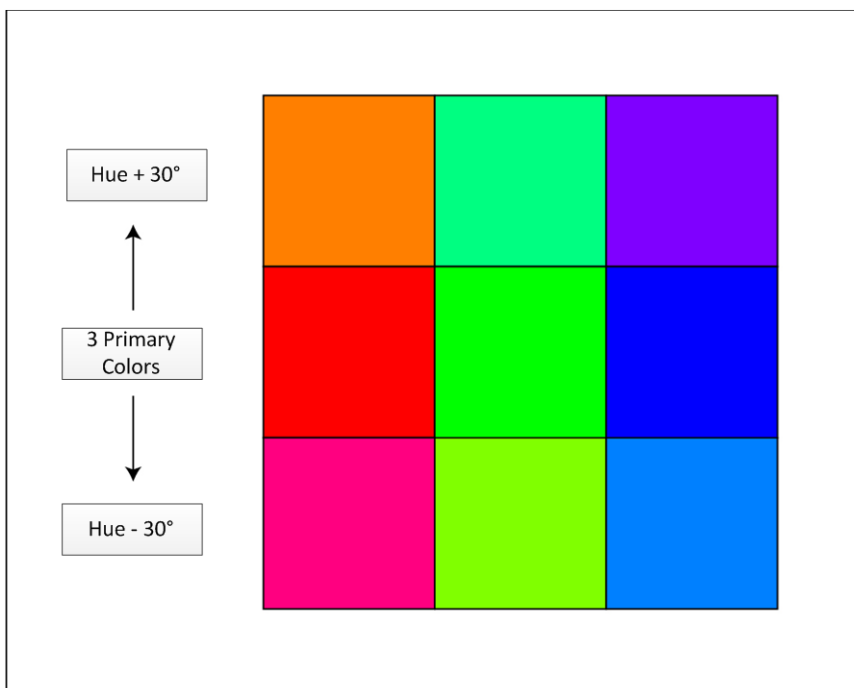
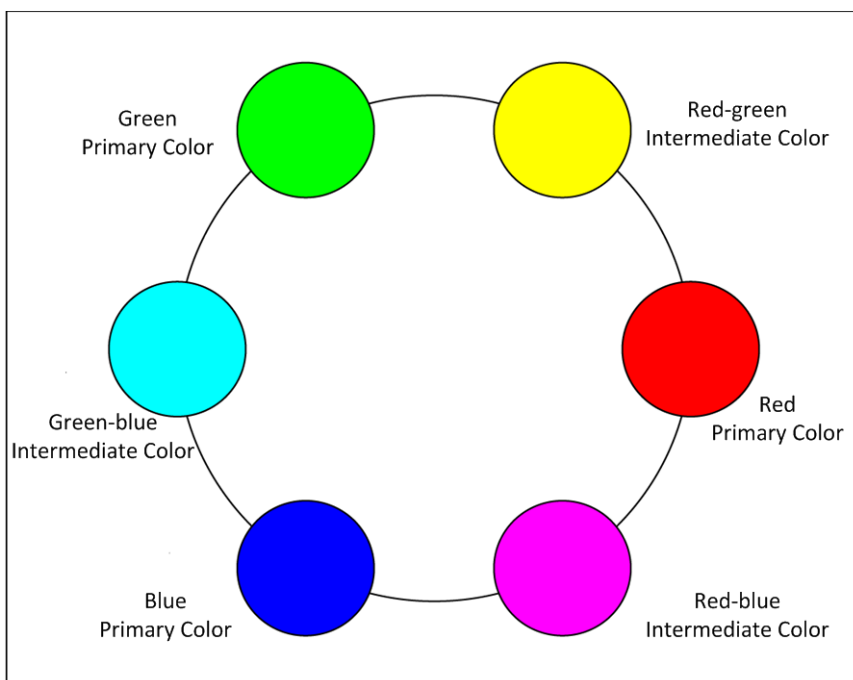
2.1.1 Setting the Input Image

Set the input image for recognizing Colorbit. The Colorbit SDK does not perform processing to detect the Colorbit position from the camera output image and to generate the input image. Use another library to perform frame recognition, detect position, and do related processing.

2.1.2 Separating Colors

The color information for the input image is separated into 4 values (red, green, blue, and achromatic). Here, the color with the lowest saturation is determined to be achromatic.

All colors with hues near red (e.g., orange or burgundy) are all determined to be red. In the same way, colors with hues near green (e.g., yellow-green) are determined to be green, and colors with hues near blue (e.g., bluish purple) are determined to be blue. Do not use the intermediate colors cyan (green + blue), magenta (red + blue), or yellow (red + green) for cell colors because their determination results can easily change according to the environment.

Figure 2-1 Example of Colors With Hues Near Red, Green, and Blue**Figure 2-2 Example of Red, Green, and Blue Intermediate Colors**

2.1.3 Extracting Colors

From the 15 points of position information for the input image, the colors separated from each point are extracted, and the corresponding Colorbit ID is returned.

Note: The Colorbit ID is fixed, and the corresponding color array is delineated in `Colorbit_list.csv`.

An ID cannot be effectively returned if there are multiple extracted points that are determined to be achromatic.

2.2 Colorbits That Are Easy to Read

Because saturation is used to extract colors, reading is easier when using color with high saturation for the 15 points to extract. Do not position the 15 points to extract on thin lines or near color boundaries.

Make sure that the image size is large enough and the image is in focus as it is actually seen through the camera.

See Chapter 3 Creating Colorbit for details on creation.

2.3 Camera Environment

Because the application uses the camera, Colorbit recognition becomes difficult if the camera environment becomes bad. The primary causes are distance from the camera to the Colorbit, ambient light brightness, and ambient light color.

2.3.1 Distance From the Camera to the Colorbit

If the distance between the camera and the Colorbit becomes too large, the size of the Colorbit in the camera image becomes small. If the image input to the Colorbit library is too small, it cannot be read, so ensure that the input image size is 64 x 64 pixels or larger.

2.3.2 Ambient Light Brightness

If the ambient light is too low, then noise becomes conspicuous, and color determination becomes difficult. Because the wrong color determination can be made due to noise, perform a read verification test and determine the usage environment for the application that uses the Colorbit SDK.

2.3.3 Ambient Light Color

With sunsets or warm-color lighting, the ambient light color takes on a red tint. When ambient light color has an effect, the saturation of other colors (blue and green when the ambient light is reddish) becomes reduced. For this reason, it may become an obstacle to color extraction. As with ambient light brightness, perform a read verification test.

3 Creating Colorbit

This chapter describes the encoding method provided to assist in Colorbit creation.

3.1 Encoding

Colorbit creation is easy with the tools provided. There are two types of encoding tools provided: writing to list and a script for Adobe Illustrator.

3.1.1 Creating with List Data

List data writes color information for an ID after that ID. The color information is 0 for red, 1 for green, and 2 for blue; these are arranged in order of the Colorbit color array. Create the code referencing this list data. See `Colorbit_list.csv` for list data.

When the ID is 128: 128,2,0,0,1,2,1,1,0,2,1,1,0,2,0,0

3.1.2 Creating with the Adobe Illustrator Script

Colorbit can be semi-automatically created by using the Adobe Illustrator script sample.

Run these scripts as shown below.

1. Create 15 paths, either rectangular or spherical.

At this time, do not set the path fill color to transparent or achromatic. See Figure 3-1.

2. Set both the path order and the Colorbit color array order to the same (the foremost is first color, the furthest back is the 15th color). See Figure 3-1.

3. Select 15, and run the script. See Figure 3-2.

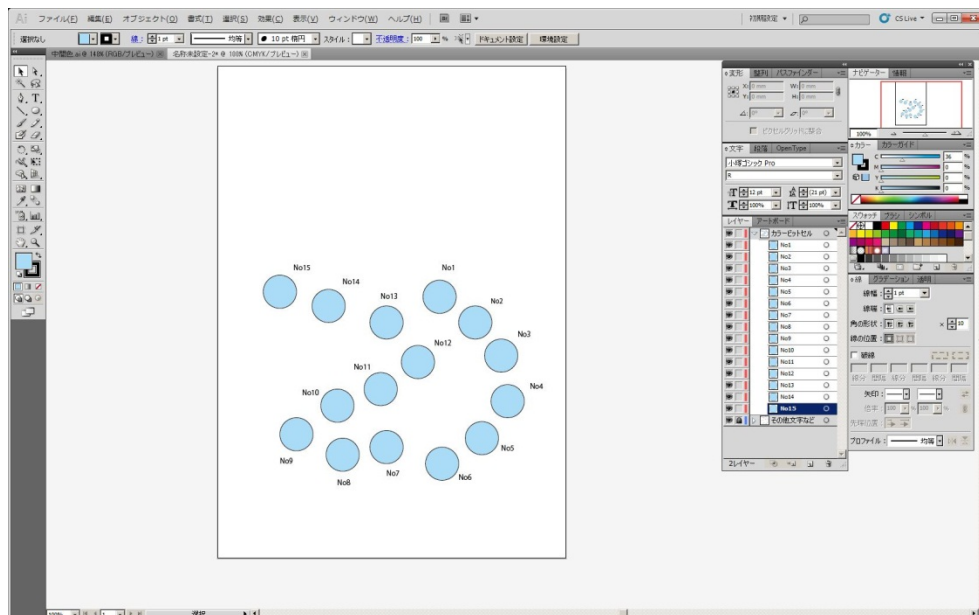
4. The ID number selection and Run button window appears, so enter the specified ID and click **Encode**. See Figure 3-3.

5. The 15 colors are automatically converted to colors that can be decoded. See Figure 3-4.

When the color mode is RGB color, use `ColorBit_encodescript_for_customcb_RGBcolor.jsx`;
when it is CMYK color, use `ColorBit_encodescript_for_customcb_CMYKcolor.jsx`.

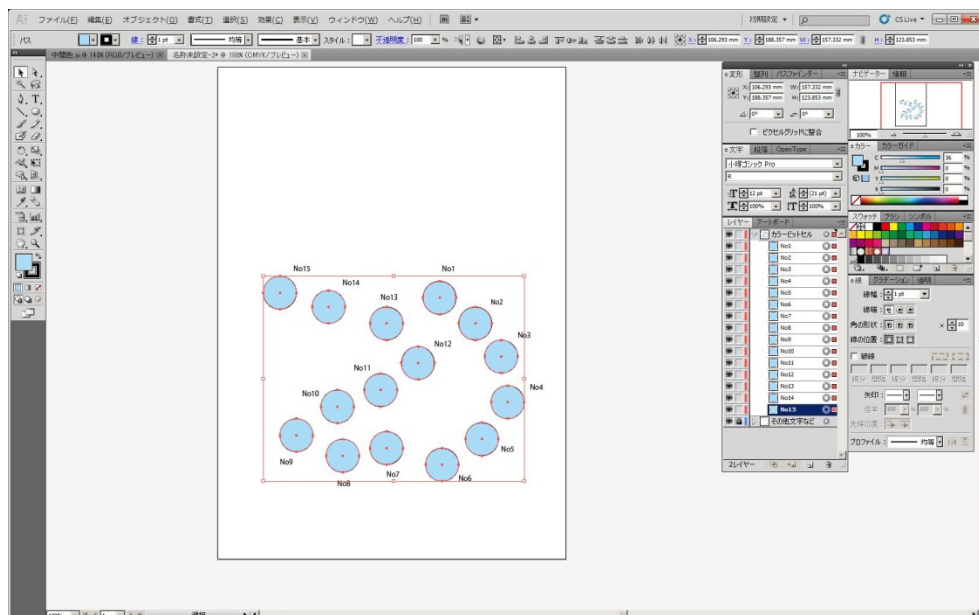
This script has been confirmed to work with Adobe Illustrator CS5 for Windows and Adobe Illustrator CS4 for Windows.

Figure 3-1 Script Execution Screen 1: Creating Paths

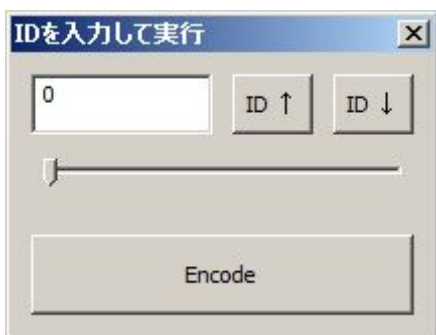


Here, a 15-point spherical path is created, and the fill is changed to a chromatic color. The condition for achromatic colors is when all CMY are 0% for CMYK colors and when all RGB values are the same for RGB color.

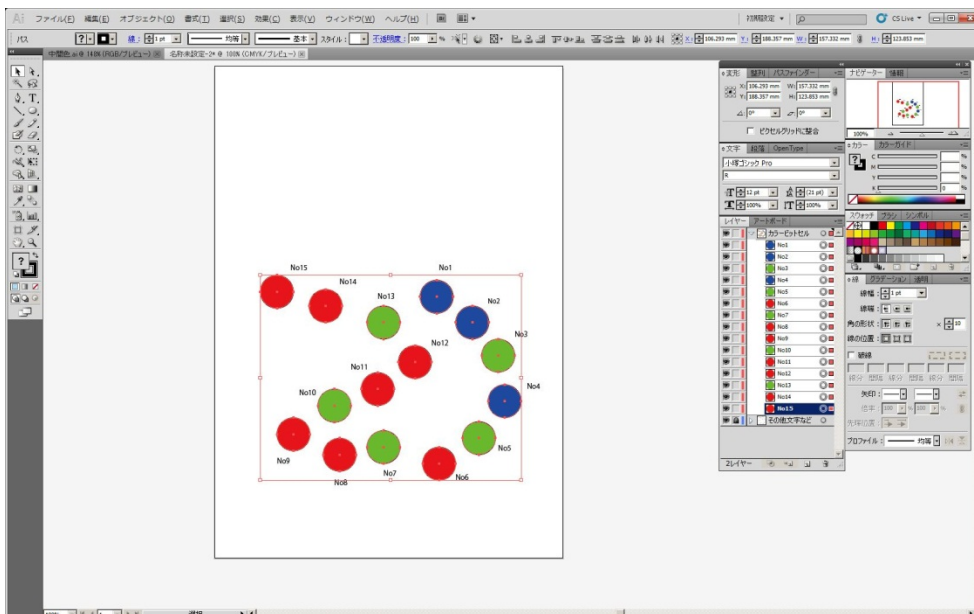
Figure 3-2 Script Execution Screen 2: Selecting Paths



Here, the created path is selected. If something other than the required path is selected and then the script runs, an error occurs.

Figure 3-3 Script Execution Screen 3: Setting IDs

Either select the script from **File > Script > Other Scripts** or drag and drop the script file to make the ID input screen appear. An error occurs here if a path is not selected or if the path is combined or grouped. Because a forced termination may occur upon occasion, we recommend saving data before running the script.

Figure 3-4 Script Execution Screen 4: Encoding Results

Set the ID and click "Encode" to have the path fill color changed automatically. The path in the forefront takes the first color of the Colorbit color array order.

3.2 Colorbit Samples

Figure 3-5 Colorbit Sample 1

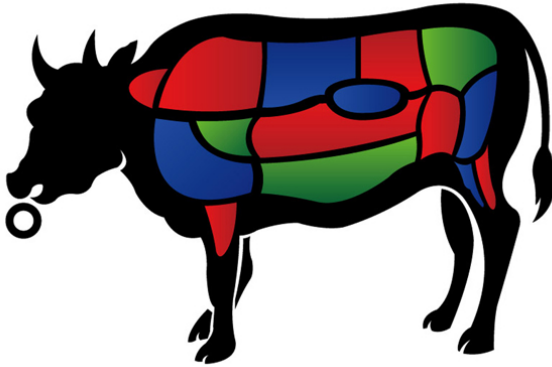


Figure 3-6 Colorbit Sample 2



Figure 3-7 Colorbit Sample 3

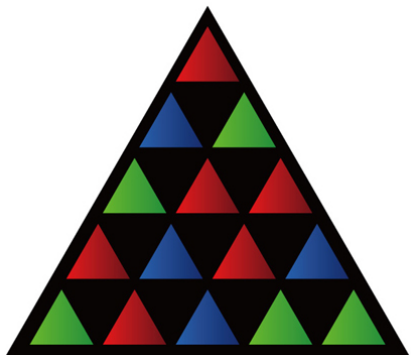


Figure 3-8 Colorbit Sample 4

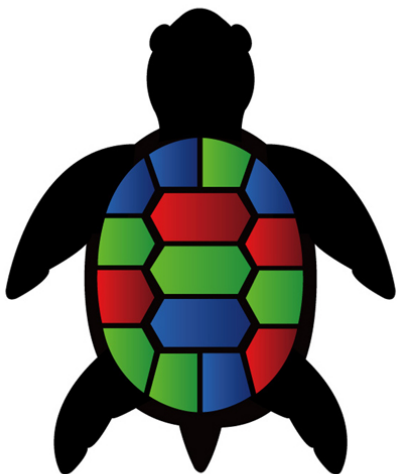


Figure 3-9 Colorbit Sample 5



Revision History

Version	Revision Date	Category	Description
1.1.3	2012/05/14	Changed	Updated the version number.
1.1.2	2012/04/04	Changed	2.1.3 Extracting Colors Noted that the Colorbit ID is fixed.
1.0.1	2012/02/28	Changed	Updated the version number.
1.0.0	2011/12/22	—	Initial version.

Colorbit is a registered trademark of B.CORE Inc.

All other company and product names in this document are trademarks or registered trademarks of their respective companies.

© 2012 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.